RSAC 2025 Conference

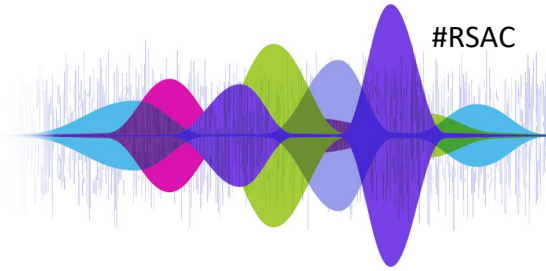
Many Voices.
One Community.

SESSION ID: SAT-W08

# Game Theory Informed Security Chaos

**Sean Atkinson**

CISO
Center for Internet Security



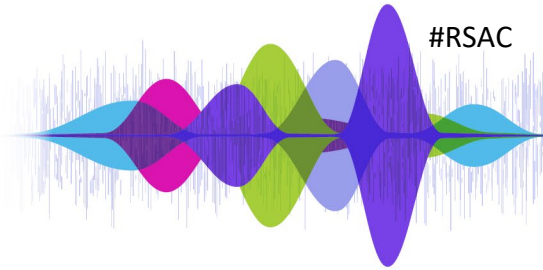
# Disclaimer

Presentations are intended for educational purposes only and do not replace independent professional judgment. Statements of fact and opinions expressed are those of the presenters individually and, unless expressly stated to the contrary, are not the opinion or position of RSA Conference LLC or any other co-sponsors. RSA Conference LLC does not endorse or approve, and assumes no responsibility for, the content, accuracy or completeness of the information presented.

Attendees should note that sessions may be audio- or video-recorded and may be published in various media, including print, audio and video formats without further notice. The presentation template and any media capture are subject to copyright protection.

© 2025 RSA Conference LLC or its affiliates. The RSAC and RSAC CONFERENCE logos and other trademarks are proprietary. All rights reserved.

# Introduction to Chaos Engineering

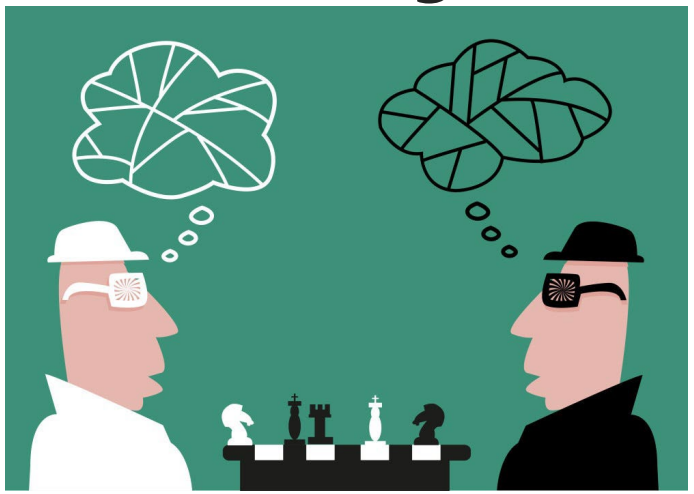


- **Definition:** Chaos Engineering is the practice of experimenting on a system to build confidence in its capability to withstand turbulent conditions.
- **Importance in Cybersecurity:** Understanding resilience through controlled adversarial conditions.
- **Key Questions:**
  - How does our system react to adversarial conditions?
  - What are the weak points?

# Why Game Theory?

- **Game Theory**: A mathematical framework for strategic interactions among rational decision-makers.
- **Application in Cybersecurity:** Models the interaction between defenders and attackers.

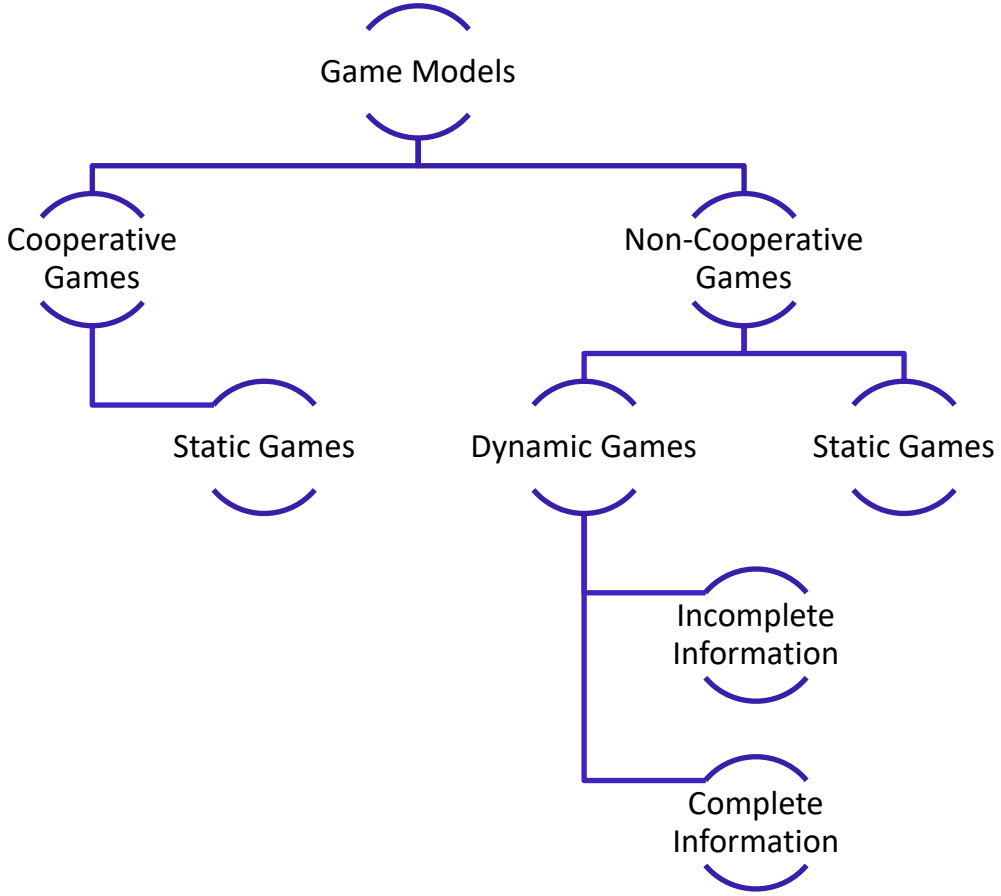
**Cybersecurity is an adversarial game—let's model it as one.**



# Cybersecurity as a Game

- **Players:** Attackers vs. Defenders
- **Strategies:** Offensive, Defensive, Adaptive Responses
- **Payoffs:** System security, data integrity, financial loss, operational disruption
- **Nash Equilibrium in Cybersecurity:** Where no player benefits from changing strategy unilaterally.

# Game Models




**Cooperative Game Theory**

**Definition**: In cooperative game theory, players can form binding agreements and coalitions to achieve better outcomes.

**Applications**: Used in scenarios like defensive resource allocation

**Non-Cooperative Game Theory**

**Definition**: In non-cooperative game theory, players make decisions independently, without forming binding agreements.

**Applications:** Attack and defense strategies, Intrusion Detection, cyber threat scenarios

# Mathematical Notation

- Let **A** be the set of attacker strategies $A = \{a_1, a_2, ..., a_n\}$
- Let **D** be the set of defender strategies $D = \{d_1, d_2, ..., d_m\}$
- Let **U_A($a_i$, $d_j$)** and **U_D($a_i$, $d_j$)** be the utility functions for attacker and defender, respectively.
- A Nash Equilibrium (stable state) occurs when:

$$U_A(a^*, d^*) \geq U_A(a, d^*) \quad \text{and} \quad U_D(a^*, d^*) \geq \quad U_D(a^*, d) \quad \forall a \in A, d \in D$$

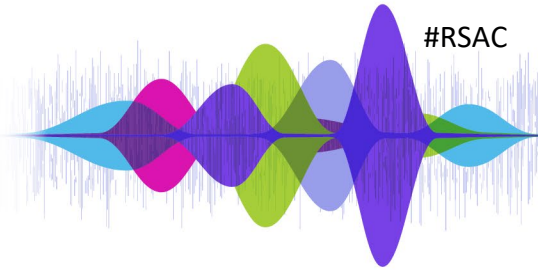
# Let's break it down!

- **UA(a∗,d∗)** This represents the utility (payoff) of the attacker when both the attacker and the defender are playing their respective optimal (equilibrium) strategies a∗ and d∗.
- **UA(a,d∗)** This represents the utility of the attacker if they deviate to some other strategy a while the defender continues playing their equilibrium strategy d∗.
- **UD(a∗,d∗)** This represents the utility (payoff) of the defender when both are playing their equilibrium strategies a∗ and d∗.
- **UD(a∗,d)** This represents the utility of the defender if they deviate to some other strategy d while the attacker continues playing their equilibrium strategy a∗.
- $\forall \mathbf{a} \in \boldsymbol{A}, \mathbf{d} \in \boldsymbol{D}$ This means that the inequality must hold for all possible alternative strategies that the attacker and defender could choose from their respective strategy sets.

# Types of Games in Cybersecurity

- **Zero-Sum Games:** One side's gain is another's loss (e.g., attacker succeeds, or defender succeeds).
- **Non-Zero-Sum Games:** Mixed incentives, where attackers and defenders may have varying levels of success/failure.
- **Sequential Games:** Decisions are made in a sequence (e.g., adaptive attack strategies, evolving defenses).
- **Bayesian Games:** Incomplete information scenarios, modeling cyber deception and intelligence gaps.

# Integrating Chaos Engineering with Game Theory



- **Traditional Chaos Engineering:** Randomized failure injection to test resilience.
- **Game-Theoretic Chaos Engineering:** Strategic failure scenarios based on attacker-defender game models.
- **Adaptive Testing:** Security teams respond dynamically, mimicking real-world attacker behavior.
- Defining a mixed strategy for the attacker (A) and defender (D)
- $\sigma \mathrm{A}(a_i) \geq 0,\ \sum i\ \sigma A(a_i) = 1\ ,\ \sigma \mathrm{A}(a_i) \geq \sum j\ \sigma D\left(d_j\right) = 1,\ \sigma D\left(d_j\right) \geq 0$

These distributions identify that a selection is random but within a rational probability for specific actions of both defender and attacker

# Integrating Chaos Engineering with Game Theory

- Expected Utility for each player: EU A($\sigma A$, $\sigma D$)

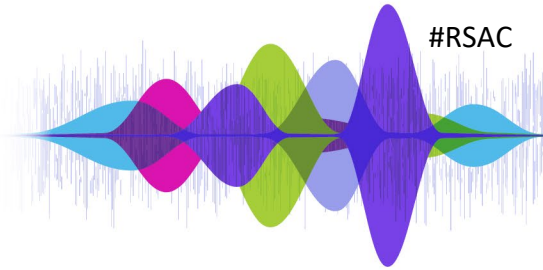
Where:

- $\sum ij\ \sigma A(a_i)\ \sigma D(d_j) UA(a_i, d_j)$ EU D($\sigma A$, $\sigma D$) =
- $\sum ij\ \sigma A(a_i)\ \sigma D(d_j) UD(a_i, d_j)$

# Symbolic representations

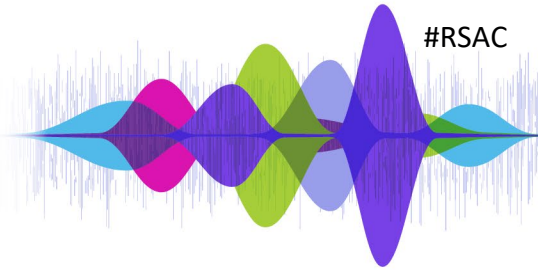
- **EU** – Expected Utility of the weighted average of the payoff from all strategies
- $\boldsymbol{\sigma A(a_i)}$ Probability of an attacker choosing strategy $(a_i)$
- $\boldsymbol{\sigma D(d_j)}$ Probability of a defender choosing strategy $(d_j)$
- $\boldsymbol{UA(a_i, d_j)}$ Utility function for the attacker choosing $(a_i)$ and defender choosing $(d_j)$
- $\boldsymbol{UD(a_i, d_j)}$ Utility function for the defender choosing $(d_j)$ and defender choosing $(a_i)$

# Designing Cyber Chaos Experiments Using Game Theory



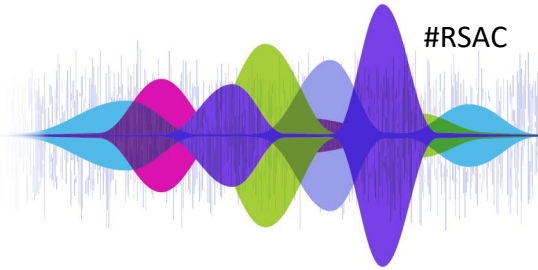
1. **Define the game:** Identify players, strategies, and payoffs.
2. **Introduce controlled failures:** Simulate security breaches and test defense responses.
3. **Observe defensive adaptations:** Monitor how teams adjust tactics.
4. **Evaluate equilibrium:** Identify optimal defensive strategies.
5. **Iterate:** Continuously refine based on results.

# Example Use Case – Ransomware Attack Simulation



- **Scenario:** Simulating a ransomware attack on a corporate network.
- **Players:** Attackers (threat actors) vs. Defenders (SOC team, IT security staff).
- **Strategies:** Attack escalation, backup restoration, network segmentation.
- **Key Learning:** What strategies yield the highest resilience?

# Example Use Case – Ransomware Attack Simulation




Game theory in modelling the game strategies

## Attacker Strategies (A):

- $a_1$ : Lauch encryption process
- $a_2$ : Perform recon before encryption
- $a_3$ : Lateral movement and privilege escalation

## Defender Strategies (D):

- $d_1$ : Early detection and network segmentation
- $d_2$ : Immutable backup storage
- $d_3$ : User training and phishing defense

# Example Use Case – Ransomware Attack Simulation



## Utility Payoff

UA($a_1$ , $d_1$) = 2 - Encryption start, but containment limits the damage

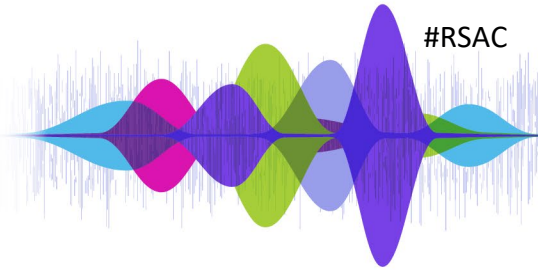
UA($a_3$ , $d_2$) = 5 - High success if Backup's are not protected

UA($a_1$ , $d_2$) = 4 - Rapid recovery mitigates impact

So, our goal in simulation is to address where A($a_i$) has the greatest impact and react. Our best response may not be enough to thwart or defend ourselves successfully from an attack.

# Example Use Case – Ransomware Attack Simulation





**Mixed Strategies** – as we have defined the approach, we will often have more than one strategy to deal with so we will need to optimize our defensive approach.

Where do we apply our resources and reduce the gaps:

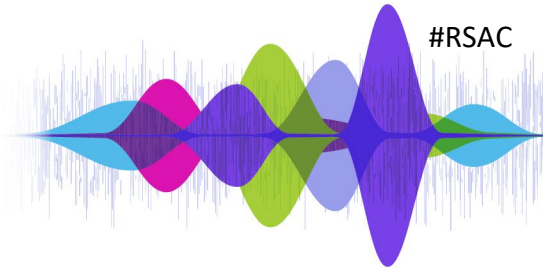
- 50% chance of investing in rapid detection $(d_1)$
- 30% chance of investing in secure backup capability $(d_2)$
- 20% chance of investing in phishing protection $(d_3)$

We will use the expected utility to address the worst-case scenario and adjust $\sigma D$ to optimize our defensive capability with a respective budget.

# Introduction to Game-Theoretic Chaos Engineering Benefits

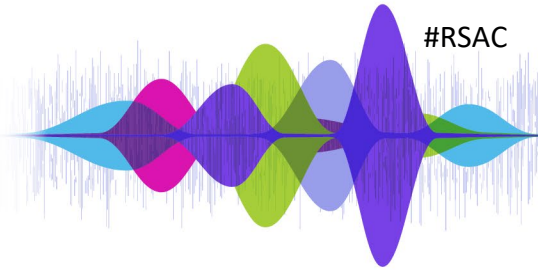
- Traditional Chaos Engineering vs. Game-Theoretic Chaos Engineering.
- Understanding how attackers think and react dynamically.
- Better modeling of real-world security incidents.

# Improved Defensive Strategies



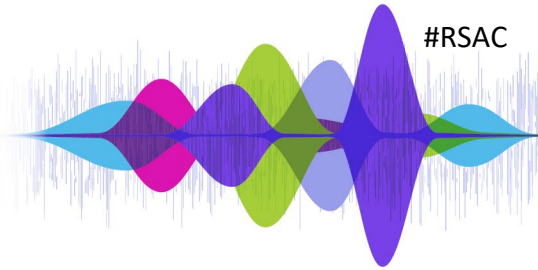
- Identifying optimal responses to different attack vectors.
  - Using specific tactics and techniques that are most likely to affect your organization to determine the application of resources and greatest Return On Security Investment (ROSI)
- Evaluating the effectiveness of mitigation strategies.
- Using mixed-strategy equilibria to inform security decisions.

# Cost-Effectiveness and Prioritization



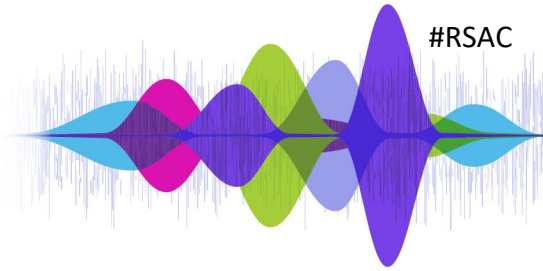
- Reducing unnecessary defensive expenditures by focusing on the most likely attack vectors.
- Game theory helps prioritize security investments based on probable attacker behaviors.
  - **Example:** Allocating resources efficiently to mitigate ransomware threats.

# Threat Intelligence and Game Theory



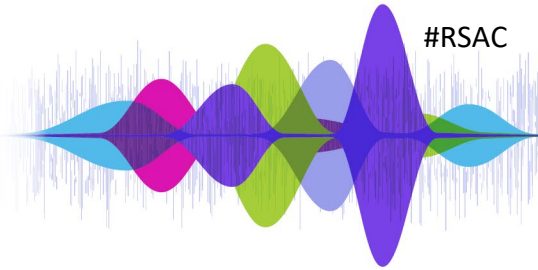
- **How Threat Intelligence Enhances Game Theory:**
  - Provides real-time data on emerging threats.
  - Helps refine attacker-defender models based on recent threat actor behaviors.
- **Use Case:**
  - Incorporating real-world threat intelligence into strategic decision-making processes.
  - Example: Mapping adversary tactics from MITRE ATT&CK framework to optimize defender responses.

# Real-World Implementation and Future Applications



- **AI-Augmented Game Theory:** Leveraging AI to simulate adversarial strategies.
- **Automated Chaos Engineering Frameworks:** Game-theory-based decision engines.
- **Cybersecurity Red vs. Blue Team Applications:** Enhancing offensive and defensive exercises.
- **Threat Intelligence Integration:** Using real-time data to refine game models.

# Apply What You Have Learned Today



- Understand your threat environment!
- Build a attack – defense tabletop strategy
- Apply a simple formulaic approach to current defensive capability and adversarial actions
- Evolve this over time and across dimensions of both adversarial ability and affected systems.

# Conclusion

- **Traditional Chaos Engineering:** Useful but lacks adversarial context.
- **Game-Theoretic Chaos Engineering:** Enhances security by modeling attacker-defender interactions.
- **Key Takeaway:**
  - By integrating strategic adversarial modeling, cybersecurity teams can improve resilience and defensive decision-making.

RSAC™ 2025 Conference

Many Voices.
**One Community.**